Bolt Beranek and Newman Inc.

(3)

LEVEL II

Report No. 4766

AD A104296

Research on Narrowband Communications

Quarterly Progress Report No. 4
18 May—17 August 1981

Prepared for:
Defense Advanced Research Projects Agency

81 9 17 001

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

14 BBN-4766

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: Report No. 4766	2. GOVT ACCESSION NO. AD-A104296
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): RESEARCH ON NARROWBAND COMMUNICATIONS	5. TYPE OF REPORT & PERIOD COVERED: Quarterly Rept No. 4, 18 May — 17 Aug 1981 (progress)
	6. PERFORMING ORG. REPORT NUMBER: BBN Report No. 4766
7. AUTHOR(s): John Makhoul, Salim Roukos, Richard Schwartz	8. CONTRACT OR GRANT NUMBER(s): F19628-80-C-0165, ARPA Order-3515
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Bolt Beranek and Newman Inc., 10 Moulton St., Cambridge, MA 02238	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS	12. REPORT DATE: August 1981
	13. NUMBER OF PAGES: 37
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Deputy for Electronic Technology (RADC/EEV), Hanscom AFB, MA 01731, Mr. Anton Segota, Contract Monitor	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this document is unlimited. It may be released to the Clearinghouse, Dept. of Commerce, for sale to the general public.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

This research was supported by the Defense Advanced Research Projects Agency under ARPA Order No. 3515, AMD. 4.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

speech compression, linear prediction, clustering, spectral template, vocoder, unsupervised learning, diphone, phonetic vocoder, phoneme recognition, time warping.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

We report on research toward a very-low-rate vocoder. This quarter we continued investigation in two areas: the phonetic vocoder and an unsupervised method for vocoding. We introduced phoneme pair probabilities to improve the accuracy of phonetic recognition. We investigated the use of the phonetic vocoder as a tool for semi-automatic labeling of speech. We also experimented with several variations of the phonetic

Cont'd

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

060100

vocoder to improve the intelligibility of the vocoded speech with a moderate increase of the bit rate. Our work in the second area concentrated on developing a novel method of spectral quantization: We quantize a sequence of spectra simultaneously. We are also evaluating a new distance metric that does not require the dynamic programming time warping.

Report No. 4766

RESEARCH ON NARROWBAND COMMUNICATIONS

Quarterly Progress Report No. 4
18 May - 17 August 1981

Prepared by:

Bolt Beranek and Newman Inc.
10 Moulton Street
Cambridge, Massachusetts 02238

Prepared for:

Defense Advanced Research Projects Agency

TABLE OF CONTENTS

1. OVERVIEW

In this Quarterly Progress Report, we present our work performed during the period May 18 to Aug. 17, 1981. Our work during the past quarter concentrated on two main areas:

1. Improvement of the phonetic vocoder and variations of the vocoder to improve intelligibility.
2. A new method for quantizing a sequence of spectra for the unsupervised method of very-low-rate vocoding.

1.1 Phonetic Vocoder

The phonetic vocoder is based on the recognition of the sequence of phonemes in the input speech. This sequence is recognized by selecting the best path through the network that matches the input. The current phoneme recognition rate of 60% is not sufficient for intelligible speech transmission.

During this quarter we investigated the usefulness of phoneme pair probabilities for improving the phoneme recognition rate. We assumed, as a first approximation, that the probability of the next phoneme depends only on the current phoneme. The probabilities were estimated from our natural speech data base. This model enhances the likelihood of phoneme sequences that correspond to the English language. In fact, the output of the

phonetic vocoder has fewer phoneme insertion errors. The recognition rate remained at 59% but the errors can be characterized as word substitution errors, i.e., the output was long strings (2 to 3 words) of either correct or wrong phonemes.

We then evaluated the performance of the current matcher in a semi-automatic labeling procedure of new speech. Basically, the operator determines the sequence of phonemes in the input. The matcher is then used to determine the location of the phonemes in the input. The results are encouraging and will speed up the labeling process.

Finally, we investigated two methods to improve the intelligibility of the vocoded speech with the current phonetic recognition algorithm, but at slightly higher bit rates than 100 b/s. The first method was to transmit, in addition to the usual information, whichever template of the diphone was actually matched. This method would increase the bit rate to around 136 b/s. In the simulation of this system the resulting speech was substantially more intelligible than that of the phonetic vocoder. The second method, with a bit rate of around 200 b/s, is a diphone vocoder. In this method, we allow any diphone to follow any diphone in the network. We then transmit the actual sequence of diphone templates matched. The output speech of this vocoder was reasonably intelligible.

1.2 Sequence Clustering

We developed earlier a Markov chain model for speech to reduce the bit rate or a vocoder by using the time dependence of quantized spectra. The bit rate was reduced from 180 b/s to 135 b/s for the spectral information, since not all spectra can follow each other in speech. While the bit rate was reduced, this method requires a large data base to estimate the necessary high order Markov model for a very-low-rate vocoder.

During this quarter, we developed a new method that uses the time dependence of spectra but does not require as much data as the Markov chain model. This new method quantizes a sequence of spectra, called a segment, simultaneously. The bit rate is reduced since not all sequences of spectra are possible in speech. We use a clustering algorithm to determine a representative set of possible segments. We also investigated an efficient new distance metric that does not require the usual dynamic programming time warping. The preliminary results during this quarter are encouraging. We plan to continue developing this approach in the coming quarter.

2. PHONETIC VOCODER

Our research on the phonetic vocoder covered four areas during the past quarter. The first area involved an initial attempt at incorporating the a priori probabilities of phoneme sequences into the distance metric. Second, we investigated the possibility of automating part of the labeling procedure for new speech. Third, we considered some variations on the basic phonetic vocoder scheme that would improve intelligibility at the cost of a small increase in average bit-rate. Finally, we consider some of the changes that might be made to the phonetic vocoder in order to improve performance.

2.1 Phoneme Sequence Probabilities

If we model the phonetic vocoder problem as a probabilistic pattern recognition task, then we can say that we would like to find the string of phonemes that are most probable, given the evidence. That is the sequence W_i for which

$$P(w_i|X) = P(w_i)\frac{P(x|w_i)}{P(x)} \qquad (1)$$

is maximum. The term $P(x|w_i)$ is the probability of the hypothesized phoneme string, w_i, producing the observed

acoustics, X. The term P(x) is the same for all theories and is, therefore, ignored.

2.1.1 Phoneme-Pair Probabilities

The term $P(w_i)$ represents the probability of one particular phoneme sequence. This is, in general, hard to estimate. However, if we define w_i as

$$w_i \equiv W_{i1}, W_{i2}, W_{i3} \cdots \quad W_{ij} \cdots W_{iN} \tag{2}$$

then

$$P(w_i) = P(W_{i1}) \cdot P(W_{i2}|W_{i1}) \cdot P(W_{i3}|W_{i2},W_{i1}) \cdot P(W_{i4}|W_{i3},W_{i2},W_{i1}) \cdot \cdots P(W_{iN}|W_{iN-1},W_{iN-2}\cdots W_{i1}) \tag{3}$$

If we assume that the probability of each phoneme is only dependent on the preceding phoneme, we can approximate $P(w_i)$ by

$$P(w_i) \simeq P(W_{i1}) \cdot \prod_{j=1}^{N} P(W_{ij}|W_{i,j-1}) \tag{4}$$

or

$$\log P(w_i) \simeq \log P(w_{i1}) + \sum_{j=1}^{N} \log P(W_{ij}|W_{i,j-1}). \tag{5}$$

Since, in the matcher, all theories are forced to start from the

silence phoneme, $P(W_{i1})$ is the same for all theories, and can, therefore, be ignored. The other terms are simply the conditional probability of each phoneme in a theory, given the preceding phoneme.

2.1.2 Probability Estimation

The data base of natural speech consists of roughly 4000 phonemes. However, only 1200 of the possible 2730 phoneme pairs occur in these 4000 samples. We clearly do not have enough samples for good estimates of all the phoneme-pair probabilities, but we can make some use of the available data. We use a "padding" procedure to avoid the case of no samples for a particular pair. That is, we assume we have seen every pair *once*. Then we add any observed samples to the data. It can be shown that, with sufficient training, the probabilities achieved by this Bayesian estimate will asymptotically approach the correct probabilities.

With so few samples, the probabilities are swamped by the padding. Therefore, we use a modified form of the padding. Rather than assuming that one sample of each pair was observed. we can assume, for instance, that 1/10 sample was observed (or alternately, multiply the observed samples by 10). This allows the probabilities to reflect the data base more quickly.

Of course, since all the probabilities used in the matcher are log-probabilities, these phoneme-pair probabilities are also converted to log-probabilities, and can be added rather than multiplied. We also allow for a weighting factor to be multiplied by each phoneme-pair log probability in order to make it comparable to the scores for the spectral match and the duration log probabilities. In our current experiments, we multiply the scores by 5.

2.1.3 Implementation

The implementation of the phoneme-pair probability score is fairly straightforward, since in the current matcher, diphone templates are never shared among different diphones. When a theory leaves a phoneme node, as soon as it gets to a spectrum node, the matcher can unambiguously determine the phoneme toward which that path will lead. Knowing the identity of the two phonemes, the matcher extracts the weighted log-probability score from a precomputed matrix and adds it to the score for that theory.

2.1.4 Results

As expected, the addition of phoneme-pair probabilities to the matcher score eliminates many unlikely phoneme sequences. In

previous versions the matcher would frequently produce several (20%) extra phonemes between the correct phonemes. These extra phonemes were mostly eliminated. The total number of correct phonemes in the output has remained essentially unchanged (59%).

Two interesting differences are apparent in the resulting phoneme sequences, as well as when we listen to the phonetic synthesis of the output. First, there are now long strings of correct phonemes. Two or three words in a row often come out perfectly. Once on the right tract, the program is fairly good at staying there. Second, there are also long strings of incorrect phonemes. These often come out as completely different, but recognizable, words from those intended. That is, the matcher behaves more like a word matcher, where the "words" are likely phoneme sequences. We have not yet determined whether this "bunching" of errors is beneficial or harmful to intelligibility.

The behavior described above is a natural consequence of the global optimality of the dynamic programming procedure and is one reason we are considering a change to an algorithm that allows us to break up these long strings. This change will be discussed in Section 2.4.3.

2.2 Semi-Automatic Labeling

One of the more time consuming aspects of this project has been that of producing a data base of labeled speech. This is one of the major reasons for the unsupervised learning approach that we are pursuing in parallel to the phonetic vocoder approach. Therefore, we decided to investigate a partially automated procedure for labeling speech.

2.2.1 Labeling Procedure

Phonetic labels, as we define them, consist of a set of phonetic text labels, each with a corresponding time marker indicating the beginning of that phonetic segment in the speech. Our normal procedure for labeling is to use a program (LPSA) that simultaneously displays several parameters of the speech waveform as a function of time (e.g., energy in different bands, formants, pitch, zerocrossing rate, etc.). As a time cursor is moved across the parameter tracks the program also displays the waveform, LPC spectrum, and parameter values at that particular time. When the user types the name of a phonetic segment, the program gives it the time corresponding to the cursor position, and displays the label under the parameters at that time. While this procedure is quite flexible and easy, it still requires about 10-15 minutes for an experienced transcriber to label a 2-second utterance.

In the semi-automatic procedure, the transcriber simply types the string of phonetic labels, without time markers, into a text file. This takes about 30 sec. for a 2 sec. utterance. This transcription is then given to the matcher as a constraint on the allowed network path. The matcher then finds the optimal match for the phonetic sequence given the diphone network. The matcher then writes a text file with the time markers and labels.

2.2.2 Labeling Results

We compared several label files produced by this semi-automatic procedure with those produced manually. Of course, the phonetic labels were the same, since they were both produced manually. The time markers were, however, slightly different. The typical difference was one or two frames. The difference was rarely three or more frames, and was never observed to be off by a whole phoneme.

Our evaluation of the performance of this automatic alignment of the labels is that they should be adjusted to match the positions that would be given by the transcriber. We feel that, since the transcriber uses a very well defined set of rules to determined the exact position of the phoneme boundaries, these rules could be incorporated in a second-pass automatic program for correcting the boundaries. This program would be allowed to

adjust each of the boundaries between two phonetic labels according to specific rules depending on the phonemes involved. For instance, the boundary between a vowel and a plosive would be positioned at the frame with the maximum drop in the energy of the preemphasized signal. The boundary between a /W/ and most vowels (except /UW/) would be positioned where the second formant was half way between the steady-state values for the /w/ and the vowel.

This two-pass alignment procedure would greatly reduce the time needed for a transcriber to label new speech. There would, of course, be a cost in terms of computer charges. The current program requires about 1-2 minutes of CPU time for a reasonably good alignment. Assuming the programs are run after midnight, the cost is about $1 for a sentence - less than the cost for the transcriber, and far less tedious.

2.2.3 Uses for Automatic Alignment

There are two primary uses for the above procedure. The first, as mentioned, is to facilitate the labeling of a large data base. A second use will be in automatic adaptation to a new speaker.

As outlined in QPR-3 [1], we have devised a method for extracting the long-term average spectrum and vocal tract length

from a short sample (10-20 sec.) of unconstrained speech. These long-term statistics, which have been successfully incorporated into our phonetic synthesis, can also be used for a gross speaker normalization for phonetic recognition. In order to perform a more accurate speaker normalization (for both recognition and synthesis) we need to extract phoneme specific features. One method that would now be possible is to require the new user to read a particular known passage. This passage would be designed so as to include a wide variety of phonemes. The passage might be anywhere from 10 sec. to 1 minute long, depending on the level of detail desired. Then, knowing the phonetic string uttered, the matcher could label this "calibration" passage. Using the labeled speech sample, phoneme specific statistics (such as spectra and duration) would be extracted. These statistics could be incorporated appropriately in the recognition and synthesis models.

We believe that an approach such as this will be necessary until we can become more sophisticated at recognizing the speech of a new speaker without any prior speaker training.

2.3 Variations to the Phonetic Vocoder

At present, the phonetic recognition rate of 60% is not sufficient for intelligible speech transmission. While we expect

the phoneme accuracy to improve to 80%, thereby improving intelligibility, we would also like to determine how much the intelligibility can be increased by the addition of a small number of extra bits. We also wanted to try to approximate the systems that would result from the unsupervised learning approach. Therefore, we simulated two variations to the basic phonetic vocoder. These variations and their corresponding results will be described below.

2.3.1 Transmit Particular Template Matched

In the phonetic vocoder, the matcher finds the closest set of diphone templates (consistent with the template recognition network) and transmits the phoneme sequence that corresponds to this diphone sequence. The phonetic synthesizer does not know which of the several diphone templates for each diphone matched best, and therefore always synthesizes one prototypical template for each diphone. At present, the template used in synthesis is the one extracted from the nonsense syllable data base. Thus, there may be a large spectral or perceptual difference between the input speech and the resynthesized speech. If the phonemes are correct, intelligibility will still be high. However, when many phonemes are wrong, intelligibility will be poor.

We could, by the additional transmission of just a few extra

bits, tell the synthesizer which of the diphone template matched best. Assuming entropy coding, this information could certainly be transmitted for an average of 2 bits per diphone. For a normal speaking rate of 12 phonemes per sec., this corresponds to 24 b/s. Since the transmitted units would no longer be as constrained, we might also require somewhat more accuracy in the duration transmitted or an additional bit specifying overall energy level. Assuming one more bit per phoneme, we get 36 b/s extra or 136 b/s total.

To simulate this system, we allowed the matcher to record on a file the exact spectral sequence that it matched for an input utterance. The pitch and voicing were taken directly from the input (instead of transmitting one value per phoneme). The gain could be either resynthesized from the templates or taken from the input. There was no smoothing of the spectra and gain between templates (as there is in phonetic synthesis). The parameter file created was then used as input to an LPC synthesizer, and speech was synthesized.

The resulting speech was substantially more intelligible than that of the phonetic vocoder. In informal listening tests several naive listeners were able to understand most of the words. When the gain was taken from the templates, there were some inappropriate loudness problems due to the fact that the

level of the "natural" templates had not been normalized. We feel that this problem can be easily corrected by the same techniques used in the phonetic synthesis.

2.3.2 Diphone Vocoder

The second simulated system was that of a diphone vocoder. In contrast to the phonetic vocoder, which transmits one phonetic unit for each one seen, the diphone vocoder can transmit arbitrary diphone templates for each diphone seen. In other words, the constraints imposed by the network are not used. The matcher is allowed to follow any diphone template with any other diphone template. This allows the matcher to model the input speech more accurately. In practice we have found that the average spectral error (euclidean distance on LARs) only decreases by 15%. In theory, the number of bits required to transmit the diphone sequence should double. However, since the templates cannot really be followed by any other (i.e., they must be roughly continuous in order to model the speech) the bit rate would go up to about 100 b/s for the templates. Since the system would now be even less like a phonetic vocoder, we feel that we would need to transmit more information for pitch, gain, voicing, and duration - probably about 100 b/s using various tricks. Thus the total bit rate might be 200 b/s.

This system was simulated by allowing the matcher to follow any diphone with any other diphone. That is, a theory coming into a phoneme node was allowed to proceed from any other phoneme node in the network. Of course, the theory stack had to be lengthened (to 4000) in order for the matcher to find a nearly optimal path. The matcher then records the particular spectra matched as described in Section 2.3.1. This parameter file is resynthesized as before.

The speech resulting from this system was reasonably intelligible. In informal tests, naive subjects typically understood all or almost all the words. More formal tests are needed.

This system is identical, in most respects, to the segment vocoder that will be described in Section 3. The only difference is that the segment inventory has been chosen by hand and represents diphone templates, rather than the result of clustering. We expect that the bit rate of this system could be lowered somewhat.

2.4 Future Changes to the Phonetic Vocoder

In this section we consider some of the changes we will implement to try to improve phoneme recognition accuracy.

2.4.1 Distance Metric

One of the major problems is the distance metric. We can say this based simply on the recognition performance. In some respect, the different samples of the same diphone all sound alike; they are all clearly understood as the same phonemes. However, in the space of the distance metric, same diphone are not closest; they are misrecognized. If we had a perfect distance metric that exactly reflected phonetic perception, then with only one, or possibly two templates for each diphone, we would get almost perfect recognition performance.

Notwithstanding the hypothetical discussion above, we would like the distance metric to reflect phonetic perception more closely. One change would be to include a term for the derivative with respect to time of the spectrum. Another change would be to use a nonlinear (e.g., Mel) frequency scale. We will also investigate the use of different spectral representations (e.g., cepstral parameters, pseudo formants, etc.). Finally, we will try to include some other features (such as energy or voicing) that are not taken into account in an LPC spectrum.

We hope that we will find a metric that, for the same amount of training, will result in higher phoneme recognition accuracy.

2.4.2 Time-Warping

The time-warping used in the current matcher is a minimum distance warping. That is, with some weak constraints, the program tries all possible warpings of the input to any given template in question, and picks the warping with the smallest spectral and duration error score. One disadvantage of this warping is that it requires a large amount of computation to consider all the alternatives. Also, it may allow time-warpings that would be perceived as making a substantial phonetic difference.

We are looking into some other time-warpings that require an order of magnitude less computation. For instance, if the warping of a segment of the input speech depended only on the input, and not on the template, then the program could perform that warping once, with no exhaustive search. The same principle could apply to the template.

It may also be more advantageous to separate the time-warping from the score for the warping. That is part of the distance between the warped input and warped template would reflect the difference in the warping.

Finally, by constraining the warping, we may actually disallow unreasonable warpings, thereby improving phoneme recognition accuracy.

We are investigating some warpings that we think will have all the properties mentioned above.

2.4.3 Pruning

The current strategy for pruning theories from the stack is to keep a fixed number of theories (say 1000) for each frame. This is called a "bounded breadth" search. The program can also use a "beam" search. In a beam search, all theories within a fixed threshold of the best scoring theory are kept. In either case, we typically find that all of the theories are only investigating the possibility of a few (about five) phonemes at any time. Said another way, there may be several hundred different alignments of the same diphone templates against the input. We would like to modify the pruning strategy so that it spreads the available theories over more phoneme possibilities.

Another change that could be important is in the details of the dynamic programming algorithm. Typically, when two alternate theories come together, we choose the better of the two and retain the score of the better theory for the one theory that will proceed. This procedure is used to find the best single path through the network. However, in the phonetic vocoder, our goal is to find the most likely phonemes at each point. To do this, we must add the two probabilities of the two theories

rather than just pick the higher one. This procedure will not find the best single path, but will tend to get more phonemes correct. It will also scatter the errors more than the dynamic programming procedure, which tends to bunch the errors.

2.4.4 Template Selection

Given that we cannot hope to find the perfect distance metric, we will put enough samples of each diphone in the network such that, given the distance metric we use, the nearest template to an unknown template will usually be one of the correct ones. As long as we can guarantee that small differences in our distance metric are not perceptible, we can always achieve this performance by increasing the training until the distance between samples of the same diphone are small.

For a large number of diphone templates, however, the storage becomes large, and the computation also grows linearly. If two templates are almost identical, they might as well be represented by a single template (with a weight or count of two). Also, we could represent several templates in a region by a single template with a "width" as big as the region. Typically, we use a mean and standard deviation for this purpose. As the number of templates gets very large this representation in terms of clusters of templates will approach the performance of the

complete set. However, for a small number (say 3) of each diphone, there may be no advantage to this clustering.

Upon examination of the matcher results we can see that the matcher almost always matches the input to the natural diphone templates rather than the nonsense diphone templates. We infer from this that the nonsense diphone templates are not typical of natural speech. We intend, therefore, to replace the nonsense diphone templates in the phonetic synthesizer with natural diphone templates. They will also be removed from each diphone on the recognition network as soon as there are a few natural templates for that diphone.

3. SEGMENT CLUSTERING

During this quarter, we started developing a new unsupervised method for reducing the bit rate of an LPC vocoder. In the previous QPR [1], we described a Markov chain model that requires a rate of 135 b/s for the spectral information only. The Markov chain model uses the statistical dependence of consecutive quantized spectra to reduce the bit rate. The new approach, called segment clustering, uses the statistical dependence of consecutive spectra by efficiently quantizing a sequence of spectra. The sequence of spectra is called a segment. We describe the new approach and present some preliminary results on the adequacy of this method for very-low-rate vocoding. We are also investigating a distance metric that involves a non-linear time warping and is computationally very efficient compared to dynamic programming time warping.

3.1 Speech Model for Segment Clustering

In the new approach, we assume that the speech has been segmented. The segmentation algorithm produces segments with a duration comparable to that of a phoneme. To reduce the bit rate, we propose to quantize the LPC spectral information in the following manner. The sequence of spectra in a segment are

quantized simultaneously by mapping the segment into a new segment called a segment template. The segment template is chosen from a set of templates by minimizing a distortion (distance) measure between the input speech segment and the template. We do not quantize pitch and gain. Hence the output speech will have the same pitch and gain tracks as the input, but the short-time spectrum will be that of the sequence of segment templates obtained by quantization. The diphone vocoder would be quite similar to this approach if the segment templates were diphones and we synthesized the matched path in the network. The success of this approach depends on finding a set of segment templates that occur frequently (or naturally) in speech. To determine such a set of templates we propose to use a segmentation algorithm to generate segments from natural speech, then to use a clustering algorithm to determine the set of "typical" segments. These "typical" segments are called segment templates. We call this method segment clustering since we are clustering segments that are sequences of spectra with variable durations. We need a distortion measure on the segments for both quantization and clustering. We point out that since the segments will usually have different durations, the distortion measure must include some form of time warping. We present in the next section a new distortion measure that includes time warping yet is computationally efficient.

3.2 Distortion Measures

In defining a distortion measure between two segments, one must specify the temporal alignment of the two segments. Fig. 1 illustrates two segments as two trajectories in spectral parameter space. We note that we only have a time sampled version (at the frame rate) of both segments. The heavy circles indicate the sampling times. One method of time alignment is linear time warping. We scale and resample the trajectories such that both have the same total duration. In isolated word recognition, a dynamic programming nonlinear time warping has been more successful. However, from looking at the two segments in Fig. 1, we can determine by ignoring the timing information, that the two segments are quite similar. They both indicate fairly closely the same sequence of parameters; however, their timing characteristics are quite different. Hence, if we transmit the timing information separately, a simple distortion measure for the segments can be defined as follows.

For each segment, we compute its total length using a Euclidean distance on LAR. Then we represent each segment by resampling the trajectory at M spatially (in the parameter space) equi-distant points. The number of points M has to be large enough so that the length of the sampling interval is small enough to capture important speech detail. The distortion

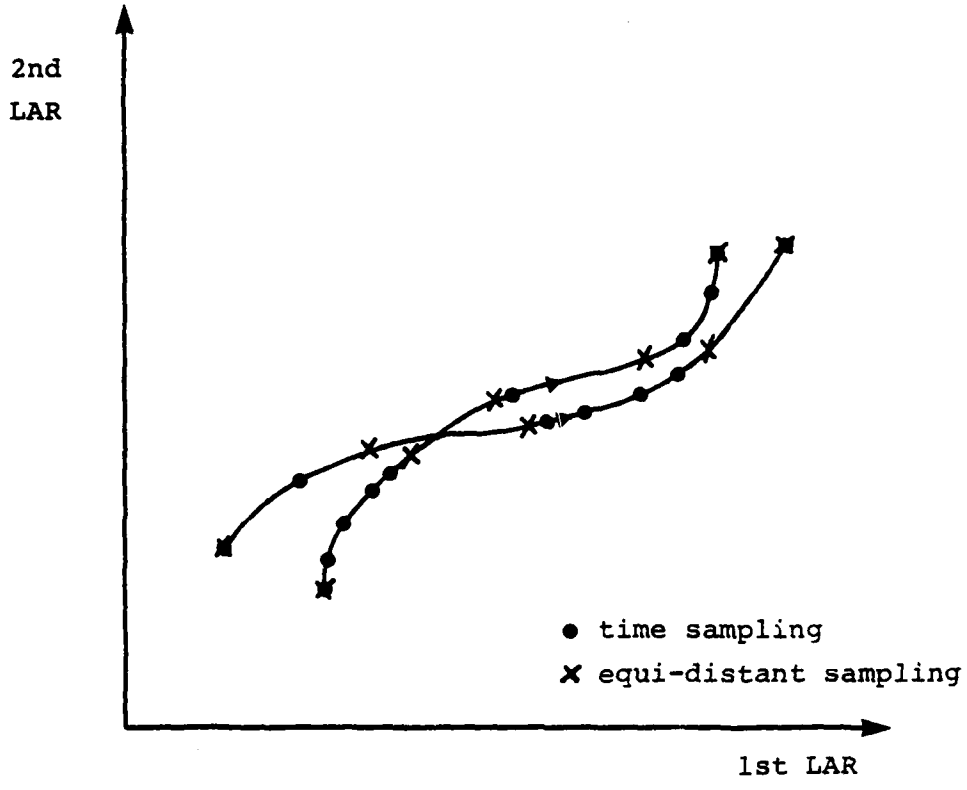

FIG. 1. Two segments in parameter space. Time is indicated by heavy circles and the arrows indicate the direction of time.

between the two segments is obtained by the sum of the Euclidean distance between the corresponding equi-distant sampling points on the two trajectories. We refer to this sampling process as spatial sampling. Using this representation, we are ignoring the timing information of the segments.

This Euclidean distortion measure is simple to implement and very efficient computationally especially when compared to a dynamic programming time alignment method.

3.3 Overview of the Proposed Vocoder

To implement a vocoder based on segment quantization, we have to solve the following two problems:

1. Determine a set of segment templates. The number of templates should be as small as possible.
2. How to quantize the input speech.

As we discussed earlier, we will use an automatic segmentation algorithm followed by a clustering algorithm to determine the templates. We will use with minor modifications the clustering algorithms developed earlier under this contract and described in QPR-1 [2] and QPR-2 [3]. We are currently developing the segmenter.

The second problem of quantizing the input speech has two

possible solutions that we plan to investigate. The first solution is to use the above segmentation algorithm and then quantize each segment using minimum distance classification as described earlier. This solution is simple and computationally efficient. However, it does not guarantee that the resulting sequence of segment templates is the best approximation to the input speech. The reason is that the segmentation is independent of the resulting segment quantization. To minimize the spectral error (distortion), we have to segment and quantize simultaneously. Basically, we need a search over all possible segmentations of the input speech to find the optimal segmentation. The resulting quantized speech will have the smallest distortion. This search will be implemented using a dynamic programming algorithm. We have implemented the first solution using hand segmentation. We describe in the next section some preliminary results using this method. We plan to implement the dynamic programming search in the near future.

3.4 Results

We have some preliminary results on the usefulness of the new distance metric and segment quantization method. To evaluate the new distance metric, we implemented an LPC vocoder to determine the required spatial sampling rate, i.e., the average number of spatial sampling points per segment.

The equi-distant spatial sampling was performed on 1 sec. speech segments. We used several values for the number of equi-distant space samples, M=10, 20, 30, 50 and 100. The timing information was transmitted at those sampling points as an additional parameter (unquantized). At the receiver, the time interval between two consecutive equi-distant sampling points is distributed uniformly along the piecewise linear trajectory. Hence in this vocoder, we stretch and compress time non-linearly between two sampling points. For M=20, we start to lose some intelligibility. At M=30, there is a slightly perceptible distortion and for M=50 and 100 there is no loss in quality.

The second experiment was an attempt to evaluate the potential intelligibility of a vocoder based on segment clustering. Instead of using clustering and segmentation to obtain the templates, we decided to use the hand labeled diphone data base as a set of segment templates for speech. We had two separate sets, the nonsense diphones and the natural diphones. We considered those separately because we thought that the natural diphones would be more typical of natural speech segments than the nonsense diphones. We note that the natural diphone data base does not have a replication of all possible diphones but has many replications of the most frequently used diphones. We used 2729 nonsense diphones and 4211 natural diphones. The average length of a nonsense diphone is 48.76 (standard

deviation=19.4) and that of a natural diphone is 40.7 (standard deviation=22). The nonsense diphones are longer in parameter space since they are usually well articulated as compared to the natural speech diphones. We used segmented speech and each segment was quantized to the nearest diphone template. We use the equi-distant space sampling representation and the Euclidean mean square distance. We used M=7 sampling points per segment. The input timing information, pitch and gain were used for the LPC synthesis. The spectrum was obtained from the sequence of chosen diphone templates.

The resulting spectral quantization error averaged over the input was 36 (mse on LAR) for the natural diphone data base. This error corresponds approximately to the error of a 5 bit single frame cluster quantizer. Further, the average number of segments in the input speech was 11 segments/sec. The bit rate for the spectral information was 12 bits x 11 segments/sec=132 bps. We used approximately 2^{12}(4096) natural diphones. The intelligibility of the vocoded speech was poor but encouraging.

In addition, the above segment quantization experiment demonstrated the following two results. First, the performance of the nonsense diphones was poor as the output speech was completely unintelligible. Second, the method was sensitive to segmentation errors. When the wrong segmentation was used, the

output speech degraded tremendously. The input speech in this experiment was hand labeled. In the future, we will use an automatic segmenter to guarantee the consistency of the segmentation for the templates and the input speech.

During the next quarter, we plan to implement the segmentation and clustering algorithms to obtain a more efficient set of segment templates than the natural diphone set, and to investigate the use of search techniques to obtain a segmentation that minimizes the spectral distortion.

REFERENCES

1. J. Makhoul, M. Krasner, S. Roukos, R. Schwartz and J. Sorensen, "Research on Narrowband Communications," Quarterly Progress Report No. 3, Contract No. F19628-80-C-0165, Bolt Beranek and Newman Inc., Tech. Report No. 4665, 18 Feb. - 17 May 1981 1981.

2. J. Makhoul, S. Roukos and R. Schwartz, "Research on Narrowband Communications," Quarterly Progress Report No. 1, Contract No. F19628-80-C-0165, Bolt Beranek and Newman Inc., Tech. Report No. 4557, 18 August - 17 November 1980.

3. J. Makhoul, M. Krasner, S. Roukos, R. Schwartz and J. Sorensen, "Research on Narrowband Communications," Quarterly Progress Report No. 2, Contract No. F19628-80-C-0165, Bolt Beranek and Newman Inc., Tech. Report No. 4620, 18 Nov. 1980 - 17 Feb. 1981.